

Les structures linéaires

1. A part les files, quelles sont les deux autres types de structures linéaires étudiées cette année ?
2. Attribuer à chaque situation décrite ci-dessous, la structure linéaire la mieux adaptée :
 - Gestion de la touche « annuler la frappe »
 - Algorithme de parcours en largeur d'un arbre
 - Tableaux de valeurs (noms, notes, villes,...)

Pour implémenter une structure de pile, on a besoin d'un nombre réduit d'opérations de bases :

- **créer_pile_vide** : fonction renvoyant une pile vide.
- **empiler** : fonction ajoutant un élément e sur une pile p , p et e étant transmis en paramètres.
- **depiler** : fonction enlevant et renvoyant un élément d'une pile transmise en paramètre.
- **est_vide** : fonction renvoyant vrai si la pile transmise en paramètre est vide, faux sinon.

3. Indiquer l'état de la file f et de la variable n après l'exécution du code suivant :

```
f = creer_file_vide()
enfiler(f, 4)
enfiler(f, 1)
enfiler(f, 3)
n = defiler(f)
enfiler(f, 8)
n = defiler(f)
```

4. On implémente une file à l'aide d'une structure de type *list*. Compléter les lignes de code nécessaires :

```
def creer_file_vide():
    '''renvoie une file vide'''
    return .....

def est_vide(f):
    '''renvoie True si la file est vide, False sinon'''
    ..... (il y a plusieurs réponses possibles, sur un
    ..... nombre de lignes variable )

def enfiler(f, x):
    '''ajoute x à la file f'''
    return f. ....

def defiler(f):
    '''enlève et renvoie le premier élément de la file'''
    assert not file_vide(f), "file vide"
    return f. ....
```